Technical Cybersecurity

Using gdb with f2

Examining f2

Very simple

- We’ll use to take a look at some common features in ELF
  - Compiler created functions
  - function prologue/epilogue
  - registers and addresses

Common GDB use

- memory reads, register contents, disassembly, etc.

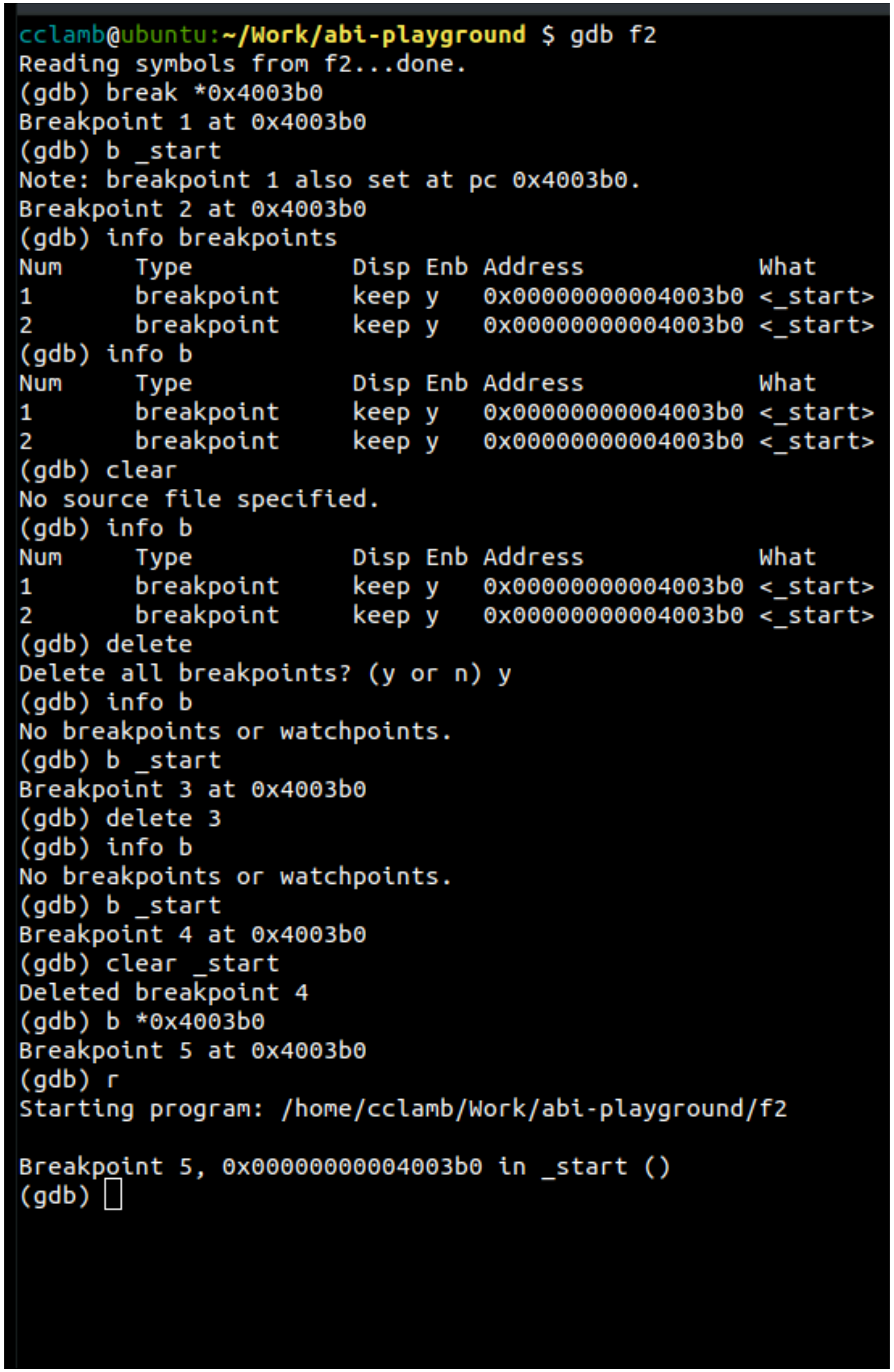
Load f2 in GDB

PREVIOUS CONFIG

- aliased gdb to gdb -q
- know the entry point from readelf -h

SET A BREAKPOINT

- multiple ways
- gdb supports shortcuts (b for break, for example)
- stop at breakpoint at program entry



```
cclamb@ubuntu:~/Work/abi-playground $ gdb f2
Reading symbols from f2...done.
(gdb) break *0x4003b0
Breakpoint 1 at 0x4003b0
(gdb) b _start
Note: breakpoint 1 also set at pc 0x4003b0.
Breakpoint 2 at 0x4003b0
(gdb) info breakpoints
Num     Type           Disp Enb Address            What
1       breakpoint     keep y   0x00000000004003b0 <_start>
2       breakpoint     keep y   0x00000000004003b0 <_start>
(gdb) info b
Num     Type           Disp Enb Address            What
1       breakpoint     keep y   0x00000000004003b0 <_start>
2       breakpoint     keep y   0x00000000004003b0 <_start>
(gdb) clear
No source file specified.
(gdb) info b
Num     Type           Disp Enb Address            What
1       breakpoint     keep y   0x00000000004003b0 <_start>
2       breakpoint     keep y   0x00000000004003b0 <_start>
(gdb) delete
Delete all breakpoints? (y or n) y
(gdb) info b
No breakpoints or watchpoints.
(gdb) b _start
Breakpoint 3 at 0x4003b0
(gdb) delete 3
(gdb) info b
No breakpoints or watchpoints.
(gdb) b _start
Breakpoint 4 at 0x4003b0
(gdb) clear _start
Deleted breakpoint 4
(gdb) b *0x4003b0
Breakpoint 5 at 0x4003b0
(gdb) r
Starting program: /home/cclamb/Work/abi-playground/f2

Breakpoint 5, 0x00000000004003b0 in _start ()
(gdb) 
```

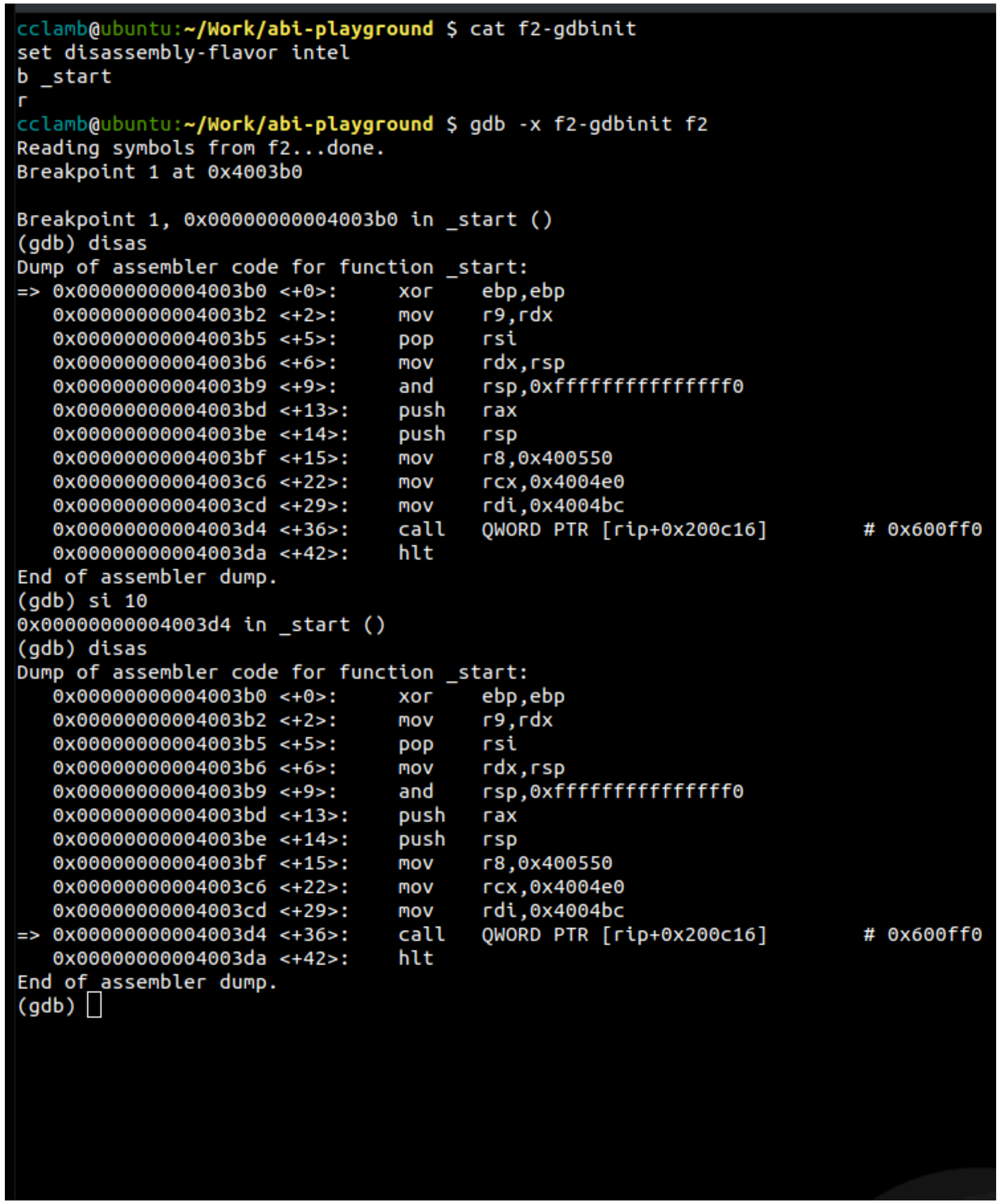
Disassemble

LOCAL GDBINIT

- I have a local gdbinit (f2-gdbinit)
- Sets context

DISASSEMBLY

- **(gdb) disas**
  - disas -> disassemble
- **(gdb) si 10**
  - si -> stepi
  - step over 10 instructions

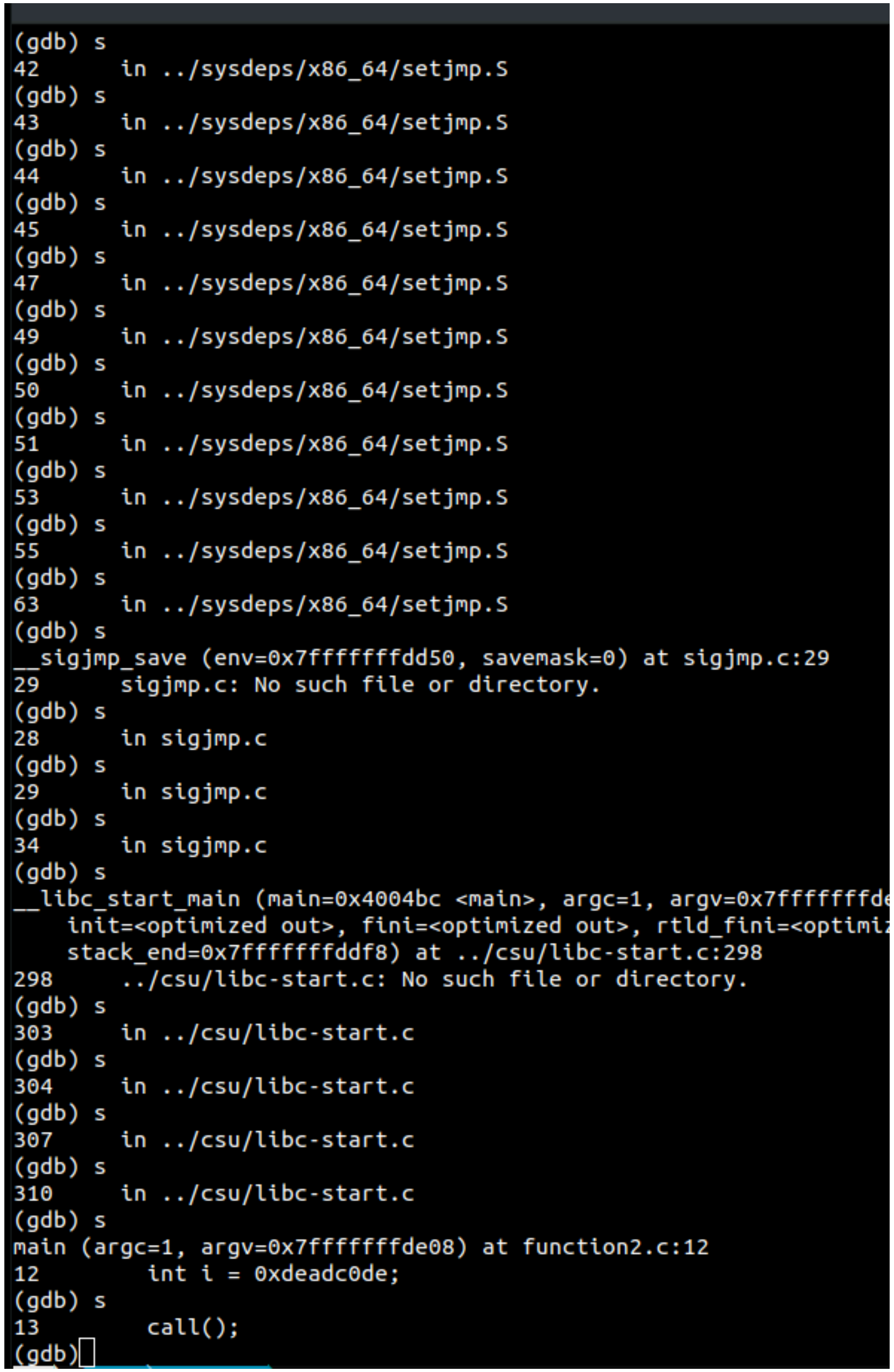


```
cclamb@ubuntu:~/Work/abi-playground $ cat f2-gdbinit
set disassembly-flavor intel
b _start
r
cclamb@ubuntu:~/Work/abi-playground $ gdb -x f2-gdbinit f2
Reading symbols from f2...done.
Breakpoint 1 at 0x4003b0

Breakpoint 1, 0x00000000004003b0 in _start ()
(gdb) disas
Dump of assembler code for function _start:
=> 0x00000000004003b0 <+0>:     xor    ebp,ebp
   0x00000000004003b2 <+2>:     mov    r9,rdx
   0x00000000004003b5 <+5>:     pop    rsi
   0x00000000004003b6 <+6>:     mov    rdx,rsp
   0x00000000004003b9 <+9>:     and    rsp,0xfffffffffffffff0
   0x00000000004003bd <+13>:    push   rax
   0x00000000004003be <+14>:    push   rsp
   0x00000000004003bf <+15>:    mov    r8,0x400550
   0x00000000004003c6 <+22>:    mov    rcx,0x4004e0
   0x00000000004003cd <+29>:    mov    rdi,0x4004bc
   0x00000000004003d4 <+36>:    call   QWORD PTR [rip+0x200c16]        # 0x600ff0
   0x00000000004003da <+42>:    hlt
End of assembler dump.
(gdb) si 10
0x00000000004003d4 in _start ()
(gdb) disas
Dump of assembler code for function _start:
   0x00000000004003b0 <+0>:     xor    ebp,ebp
   0x00000000004003b2 <+2>:     mov    r9,rdx
   0x00000000004003b5 <+5>:     pop    rsi
   0x00000000004003b6 <+6>:     mov    rdx,rsp
   0x00000000004003b9 <+9>:     and    rsp,0xfffffffffffffff0
   0x00000000004003bd <+13>:    push   rax
   0x00000000004003be <+14>:    push   rsp
   0x00000000004003bf <+15>:    mov    r8,0x400550
   0x00000000004003c6 <+22>:    mov    rcx,0x4004e0
   0x00000000004003cd <+29>:    mov    rdi,0x4004bc
=> 0x00000000004003d4 <+36>:    call   QWORD PTR [rip+0x200c16]        # 0x600ff0
   0x00000000004003da <+42>:    hlt
End of assembler dump.
(gdb) 
```

Tracing

AFTER _START

- libc-start.c
- cxa_atexit.c
- setjmp.S
- sigjmp.c
- …then into your main()!

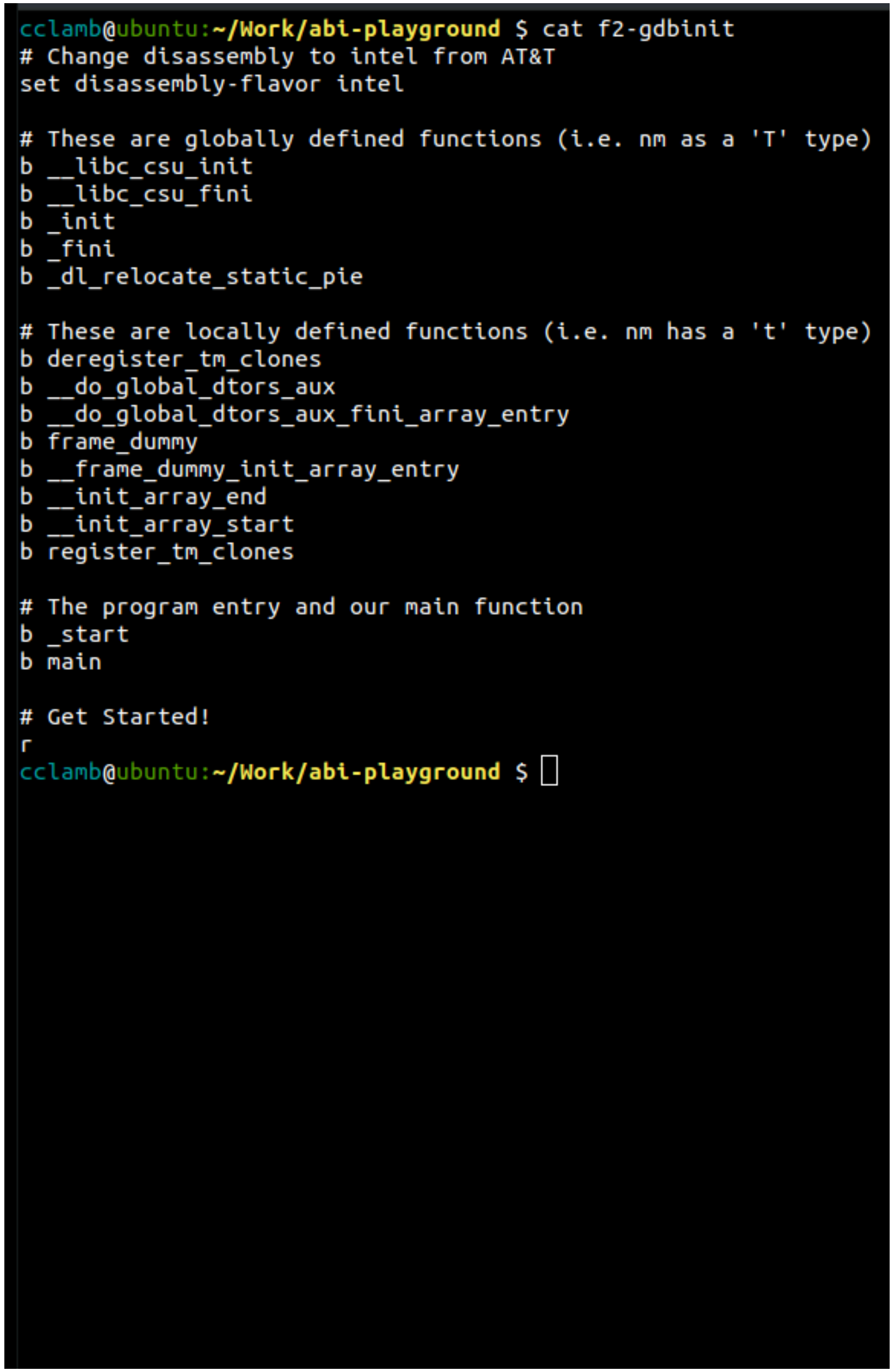


```
(gdb) s
42      in ../sysdeps/x86_64/setjmp.S
(gdb) s
43      in ../sysdeps/x86_64/setjmp.S
(gdb) s
44      in ../sysdeps/x86_64/setjmp.S
(gdb) s
45      in ../sysdeps/x86_64/setjmp.S
(gdb) s
47      in ../sysdeps/x86_64/setjmp.S
(gdb) s
49      in ../sysdeps/x86_64/setjmp.S
(gdb) s
50      in ../sysdeps/x86_64/setjmp.S
(gdb) s
51      in ../sysdeps/x86_64/setjmp.S
(gdb) s
53      in ../sysdeps/x86_64/setjmp.S
(gdb) s
55      in ../sysdeps/x86_64/setjmp.S
(gdb) s
63      in ../sysdeps/x86_64/setjmp.S
(gdb) s
__sigjmp_save (env=0x7fffffffdd50, savemask=0) at sigjmp.c:29
29      sigjmp.c: No such file or directory.
(gdb) s
28      in sigjmp.c
(gdb) s
29      in sigjmp.c
(gdb) s
34      in sigjmp.c
(gdb) s
__libc_start_main (main=0x4004bc <main>, argc=1, argv=0x7fffffffde
    init=<optimized out>, fini=<optimized out>, rtld_fini=<optimiz
    stack_end=0x7fffffffddf8) at ../csu/libc-start.c:298
298     ../csu/libc-start.c: No such file or directory.
(gdb) s
303     in ../csu/libc-start.c
(gdb) s
304     in ../csu/libc-start.c
(gdb) s
307     in ../csu/libc-start.c
(gdb) s
310     in ../csu/libc-start.c
(gdb) s
main (argc=1, argv=0x7fffffffde08) at function2.c:12
12        int i = 0xdeadc0de;
(gdb) s
13        call();
(gdb)
```

Moar Tracing!

EDIT F2-GDBINIT

- Keep disassembly flavor
- add lots of breakpoints
  - Global functions
  - Local functions
  - Entry point
  - main(.)

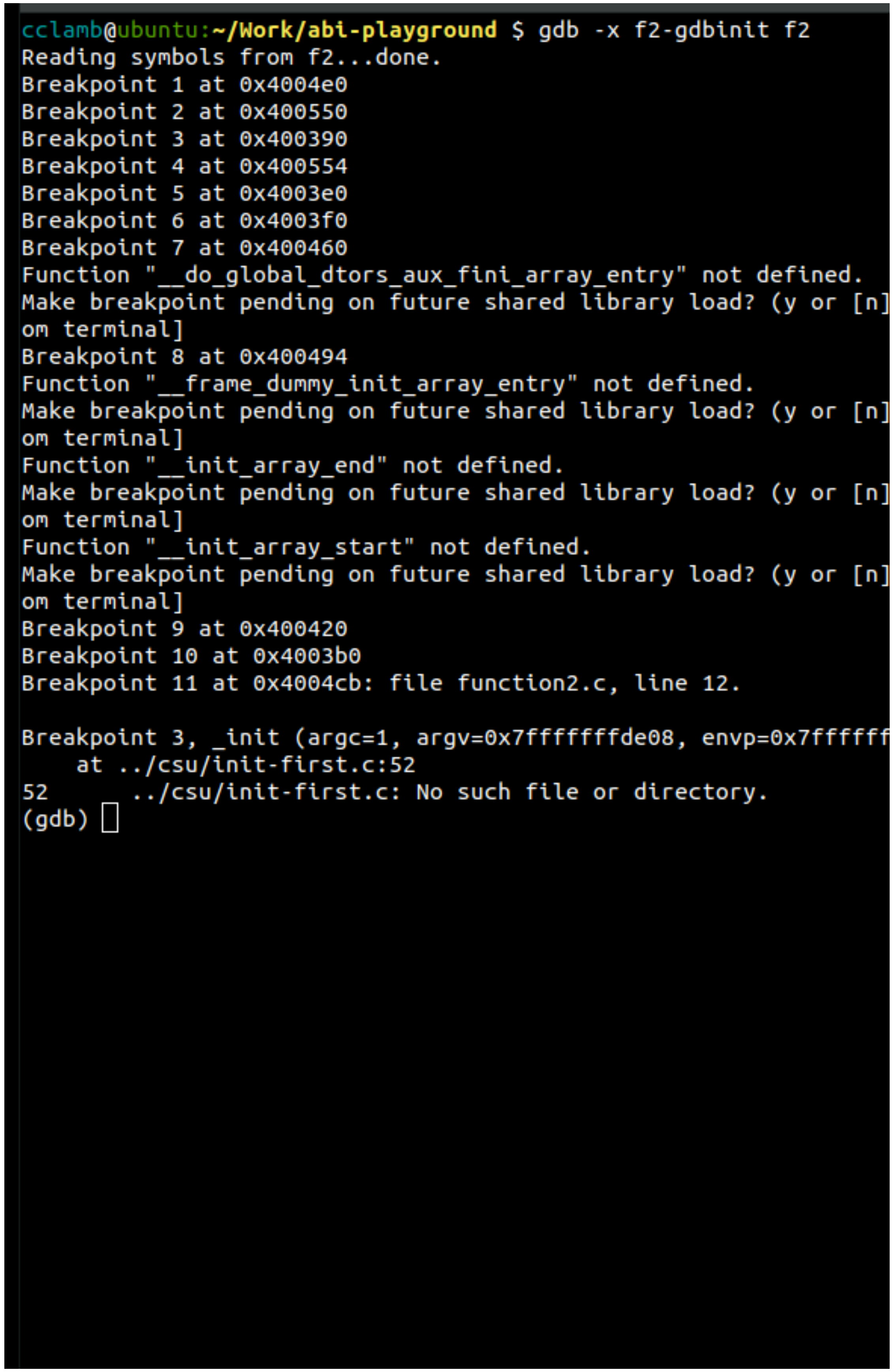


```
cclamb@ubuntu:~/Work/abi-playground $ cat f2-gdbinit
# Change disassembly to intel from AT&T
set disassembly-flavor intel

# These are globally defined functions (i.e. nm as a 'T' type)
b __libc_csu_init
b __libc_csu_fini
b _init
b _fini
b _dl_relocate_static_pie

# These are locally defined functions (i.e. nm has a 't' type)
b deregister_tm_clones
b __do_global_dtors_aux
b __do_global_dtors_aux_fini_array_entry
b frame_dummy
b __frame_dummy_init_array_entry
b __init_array_end
b __init_array_start
b register_tm_clones

# The program entry and our main function
b _start
b main

# Get Started!
r
cclamb@ubuntu:~/Work/abi-playground $ 
```

What happens?

START UP GDB

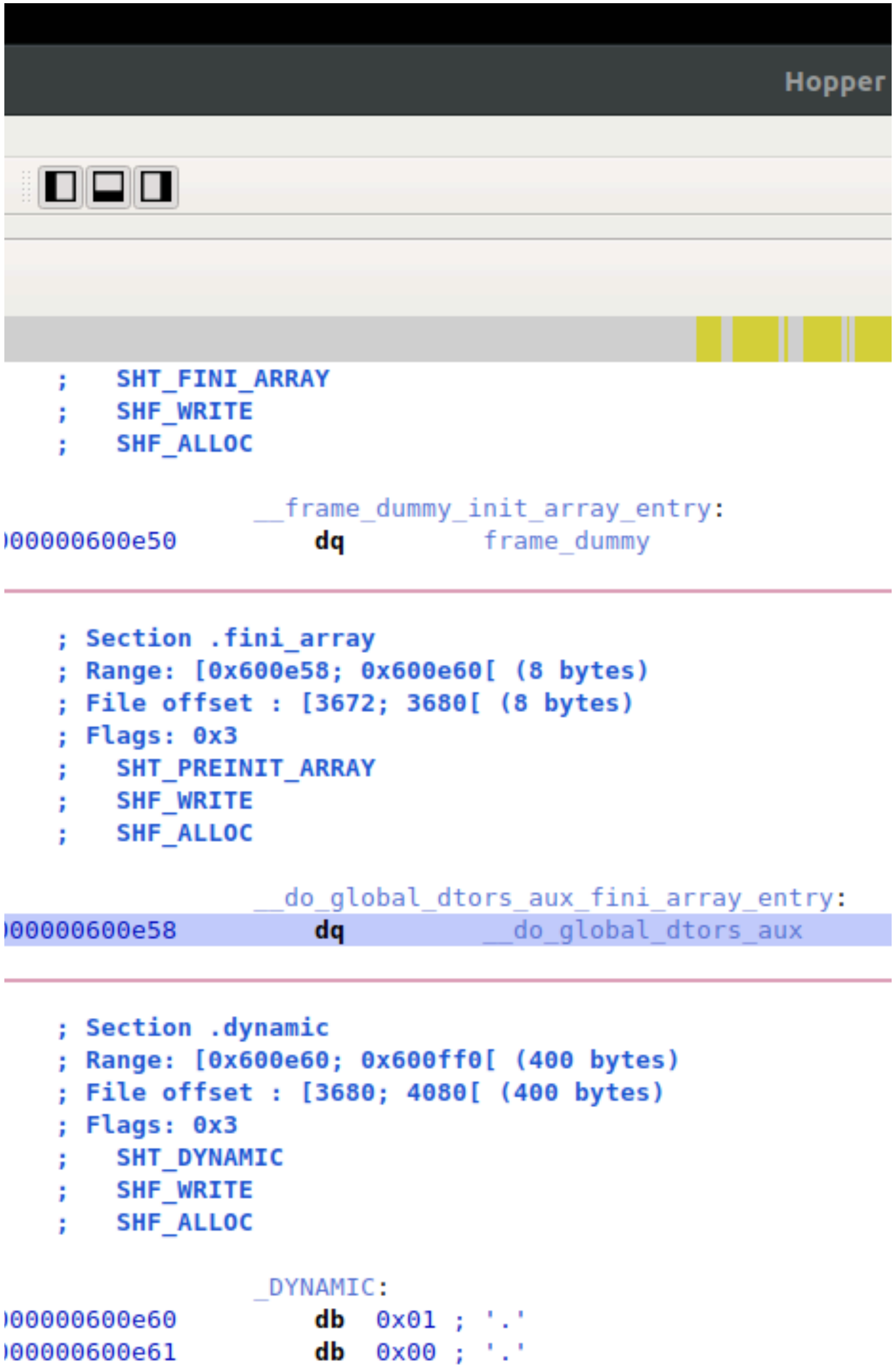
- Most of the breakpoints work
- Some don't!
    - …they're not defined?
    - Let's look at them



```
cclamb@ubuntu:~/Work/abi-playground $ gdb -x f2-gdbinit f2
Reading symbols from f2...done.
Breakpoint 1 at 0x4004e0
Breakpoint 2 at 0x400550
Breakpoint 3 at 0x400390
Breakpoint 4 at 0x400554
Breakpoint 5 at 0x4003e0
Breakpoint 6 at 0x4003f0
Breakpoint 7 at 0x400460
Function "__do_global_dtors_aux_fini_array_entry" not defined.
Make breakpoint pending on future shared library load? (y or [n]
om terminal]
Breakpoint 8 at 0x400494
Function "__frame_dummy_init_array_entry" not defined.
Make breakpoint pending on future shared library load? (y or [n]
om terminal]
Function "__init_array_end" not defined.
Make breakpoint pending on future shared library load? (y or [n]
om terminal]
Function "__init_array_start" not defined.
Make breakpoint pending on future shared library load? (y or [n]
om terminal]
Breakpoint 9 at 0x400420
Breakpoint 10 at 0x4003b0
Breakpoint 11 at 0x4004cb: file function2.c, line 12.

Breakpoint 3, _init (argc=1, argv=0x7fffffffde08, envp=0x7ffffff
    at ../csu/init-first.c:52
52      ../csu/init-first.c: No such file or directory.
(gdb) 
```

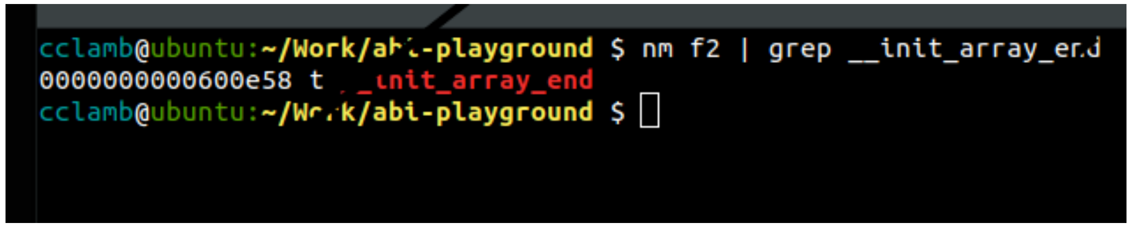
Examination

What's there?

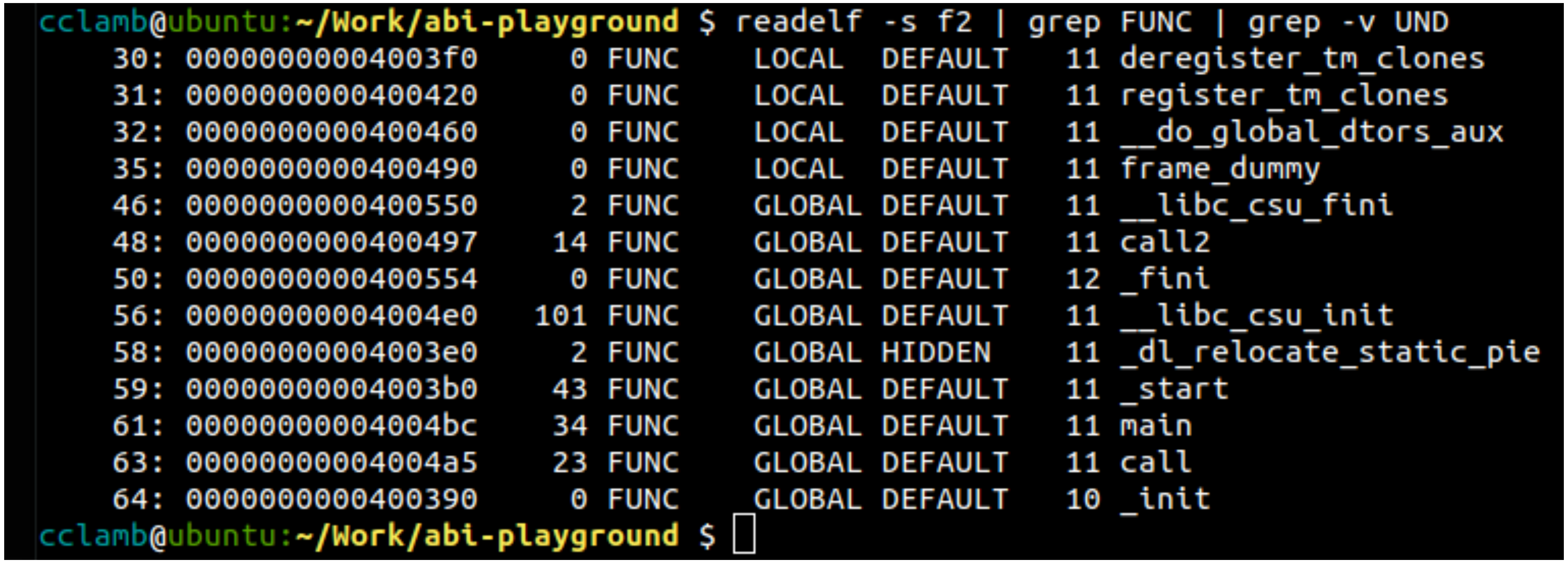


```
;    SHT_FINI_ARRAY
;    SHF_WRITE
;    SHF_ALLOC

                    __frame_dummy_init_array_entry:
00000600e50         dq          frame_dummy

; Section .fini_array
; Range: [0x600e58; 0x600e60[ (8 bytes)
; File offset : [3672; 3680[ (8 bytes)
; Flags: 0x3
;    SHT_PREINIT_ARRAY
;    SHF_WRITE
;    SHF_ALLOC

                    __do_global_dtors_aux_fini_array_entry:
00000600e58         dq          __do_global_dtors_aux

; Section .dynamic
; Range: [0x600e60; 0x600ff0[ (400 bytes)
; File offset : [3680; 4080[ (400 bytes)
; Flags: 0x3
;    SHT_DYNAMIC
;    SHF_WRITE
;    SHF_ALLOC

                    _DYNAMIC:
00000600e60         db   0x01 ; '.'
00000600e61         db   0x00 ; '.'
```

Get the address



```
cclamb@ubuntu:~/Work/abi-playground $ nm f2 | grep __init_array_end
0000000000600e58 t __init_array_end
cclamb@ubuntu:~/Work/abi-playground $ 
```



```
cclamb@ubuntu:~/Work/abi-playground $ readelf -s f2 | grep FUNC | grep -v UND
    30: 00000000004003f0     0 FUNC    LOCAL  DEFAULT   11 deregister_tm_clones
    31: 0000000000400420     0 FUNC    LOCAL  DEFAULT   11 register_tm_clones
    32: 0000000000400460     0 FUNC    LOCAL  DEFAULT   11 __do_global_dtors_aux
    35: 0000000000400490     0 FUNC    LOCAL  DEFAULT   11 frame_dummy
    46: 0000000000400550     2 FUNC    GLOBAL DEFAULT   11 __libc_csu_fini
    48: 0000000000400497    14 FUNC    GLOBAL DEFAULT   11 call2
    50: 0000000000400554     0 FUNC    GLOBAL DEFAULT   12 _fini
    56: 00000000004004e0   101 FUNC    GLOBAL DEFAULT   11 __libc_csu_init
    58: 00000000004003e0     2 FUNC    GLOBAL HIDDEN    11 _dl_relocate_static_pie
    59: 00000000004003b0    43 FUNC    GLOBAL DEFAULT   11 _start
    61: 00000000004004bc    34 FUNC    GLOBAL DEFAULT   11 main
    63: 00000000004004a5    23 FUNC    GLOBAL DEFAULT   11 call
    64: 0000000000400390     0 FUNC    GLOBAL DEFAULT   10 _init
cclamb@ubuntu:~/Work/abi-playground $ 
```

Choose Carefully!

Not all commands created equal

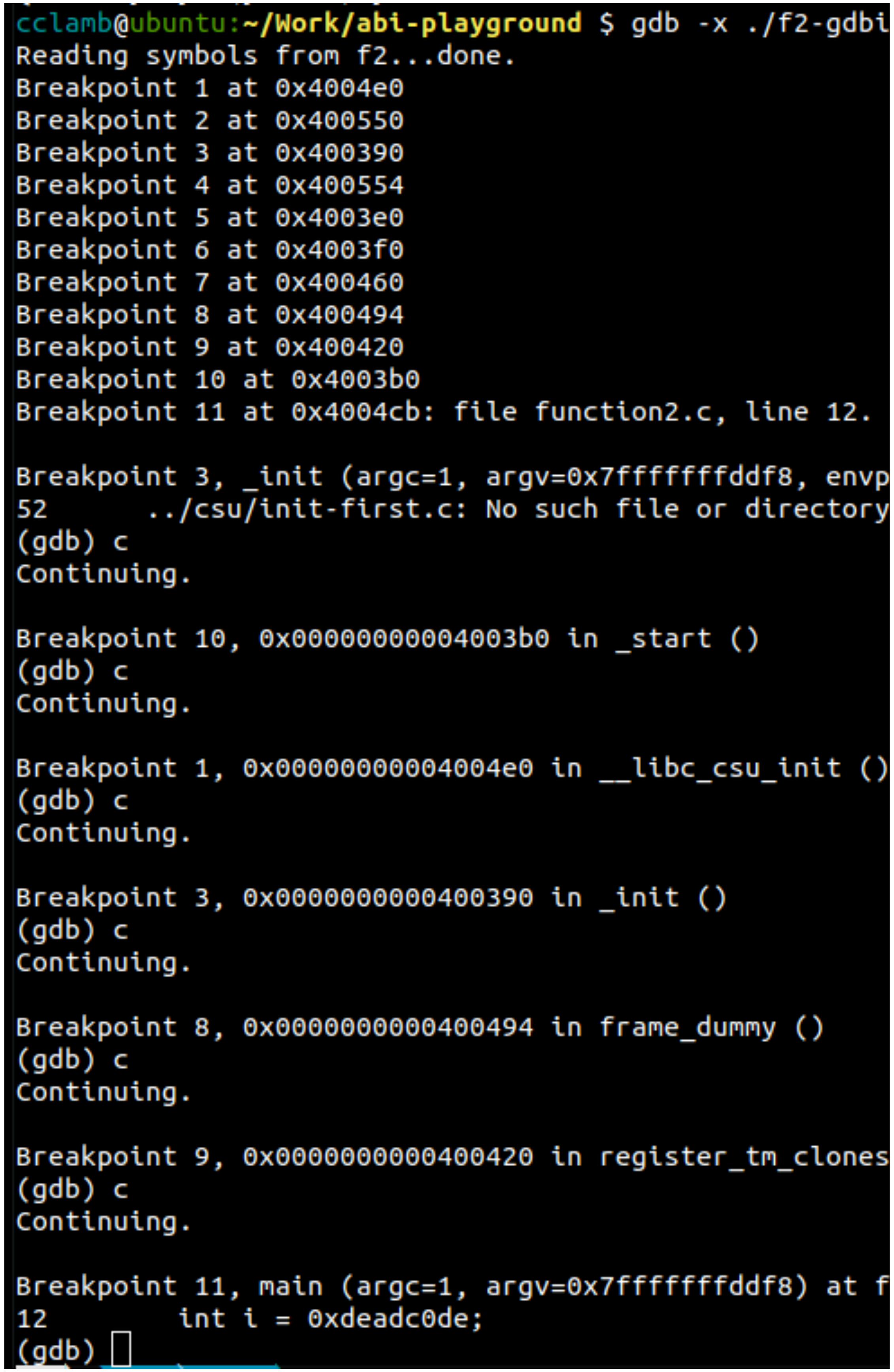
Trace again

We can see calls

- This is everything that's called before a program runs.
- All you did was write main(.)!

More GDB commands

- c -> continue
    - continues execution after a breakpoint



```
cclamb@ubuntu:~/Work/abi-playground $ gdb -x ./f2-gdbi
Reading symbols from f2...done.
Breakpoint 1 at 0x4004e0
Breakpoint 2 at 0x400550
Breakpoint 3 at 0x400390
Breakpoint 4 at 0x400554
Breakpoint 5 at 0x4003e0
Breakpoint 6 at 0x4003f0
Breakpoint 7 at 0x400460
Breakpoint 8 at 0x400494
Breakpoint 9 at 0x400420
Breakpoint 10 at 0x4003b0
Breakpoint 11 at 0x4004cb: file function2.c, line 12.

Breakpoint 3, _init (argc=1, argv=0x7fffffffddf8, envp
52      ../csu/init-first.c: No such file or directory
(gdb) c
Continuing.

Breakpoint 10, 0x00000000004003b0 in _start ()
(gdb) c
Continuing.

Breakpoint 1, 0x00000000004004e0 in __libc_csu_init ()
(gdb) c
Continuing.

Breakpoint 3, 0x0000000000400390 in _init ()
(gdb) c
Continuing.

Breakpoint 8, 0x0000000000400494 in frame_dummy ()
(gdb) c
Continuing.

Breakpoint 9, 0x0000000000400420 in register_tm_clones
(gdb) c
Continuing.

Breakpoint 11, main (argc=1, argv=0x7fffffffddf8) at f
12        int i = 0xdeadc0de;
(gdb) 
```

We’re going to continue with f2.